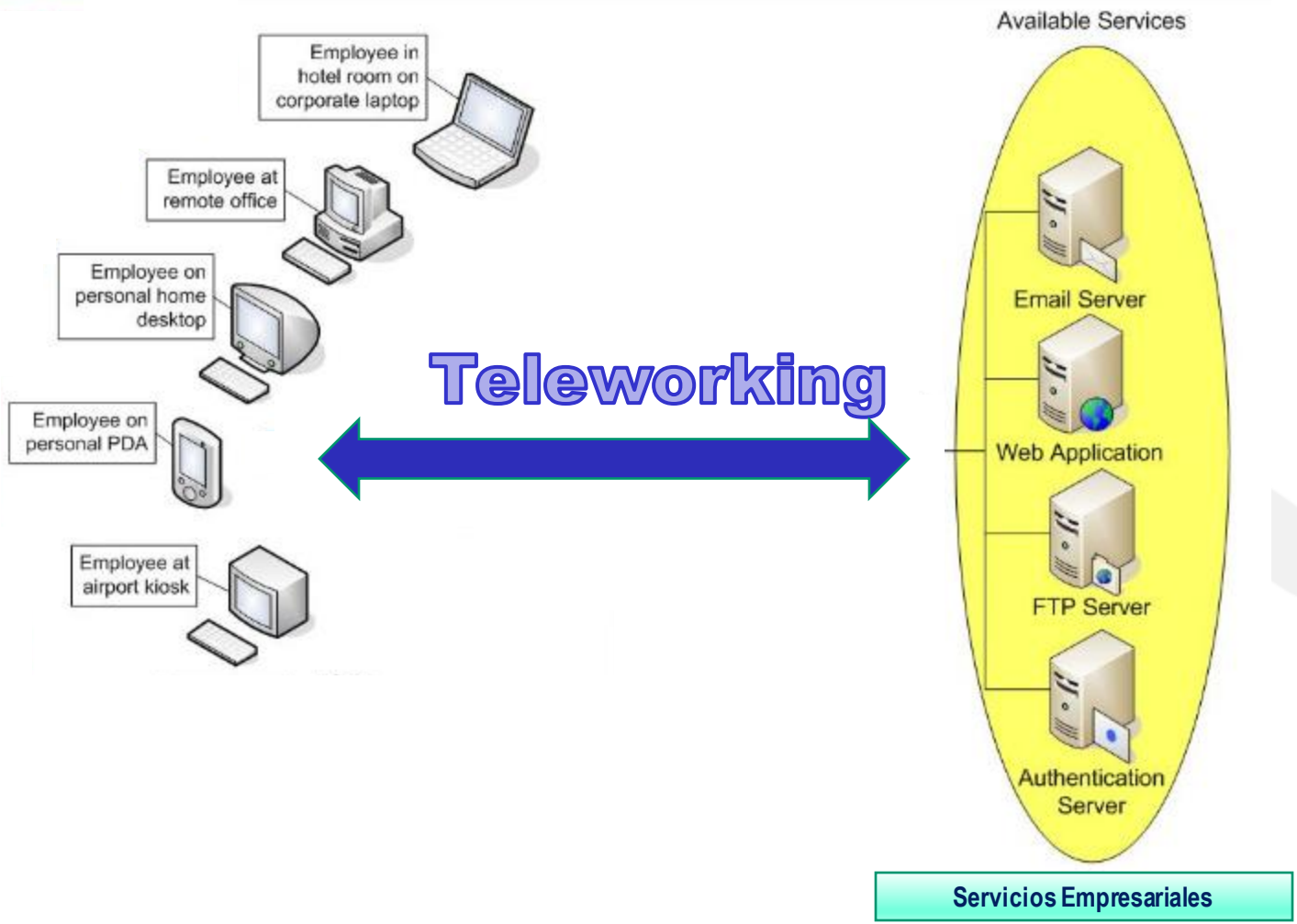
PAI-3. BYODSEC-BRING YOUR OWN DEVICE SEGURO PARA UNA UNIVERSIDAD PÚBLICA USANDO ROAD WARRIOR VPN SSL

Ángel Jesús Varela Vaca
Grupo de Investigación **IDEA Research Group**,
Universidad de Sevilla

Acceso Remoto – Road Warrior



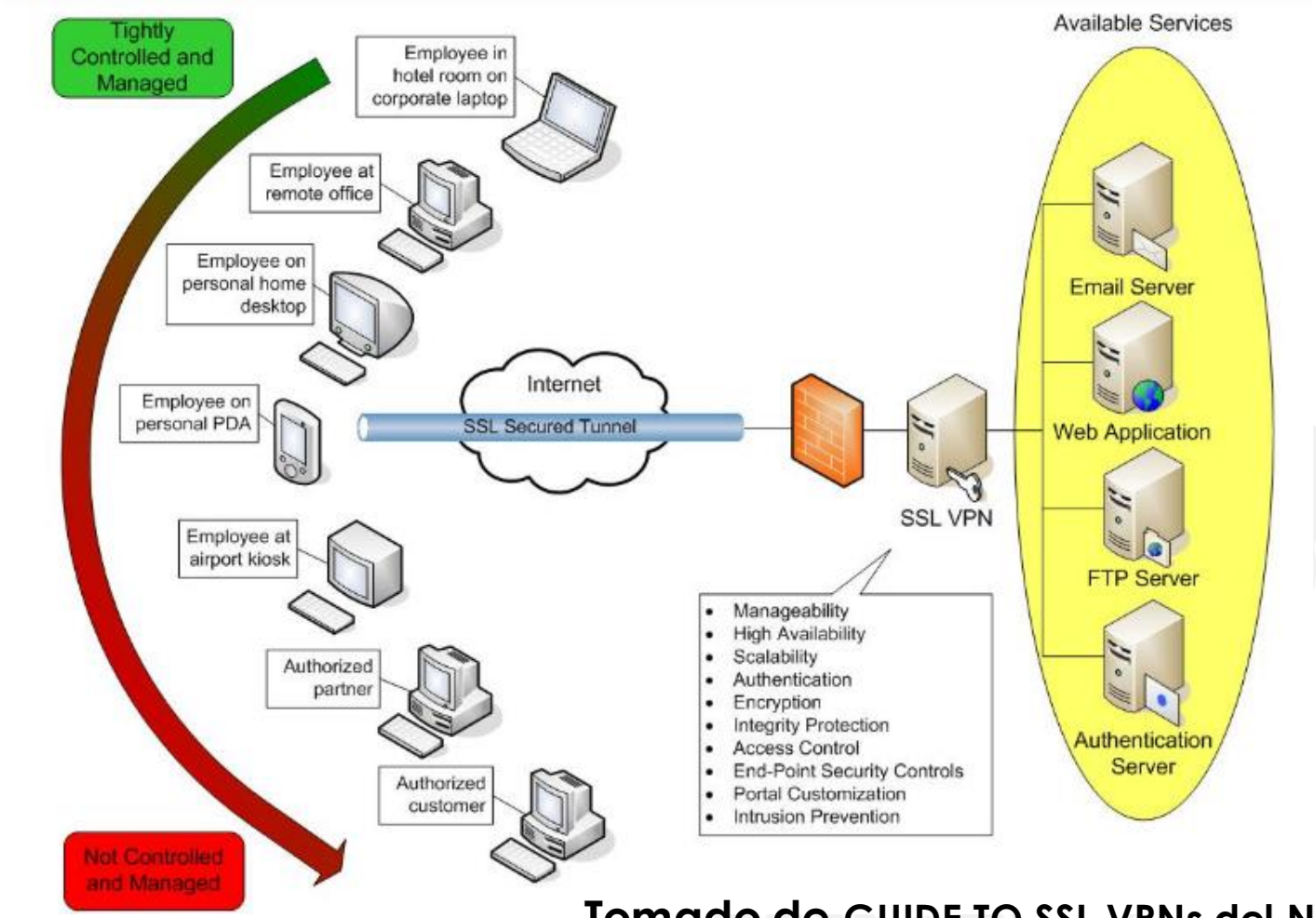
Bring your Own Device (BYOD)

"… deberá ser confidenciales, íntegras y además autenticadas"

Open Security Architecture (OSA) -
SC-09 Transmision confidentiality:

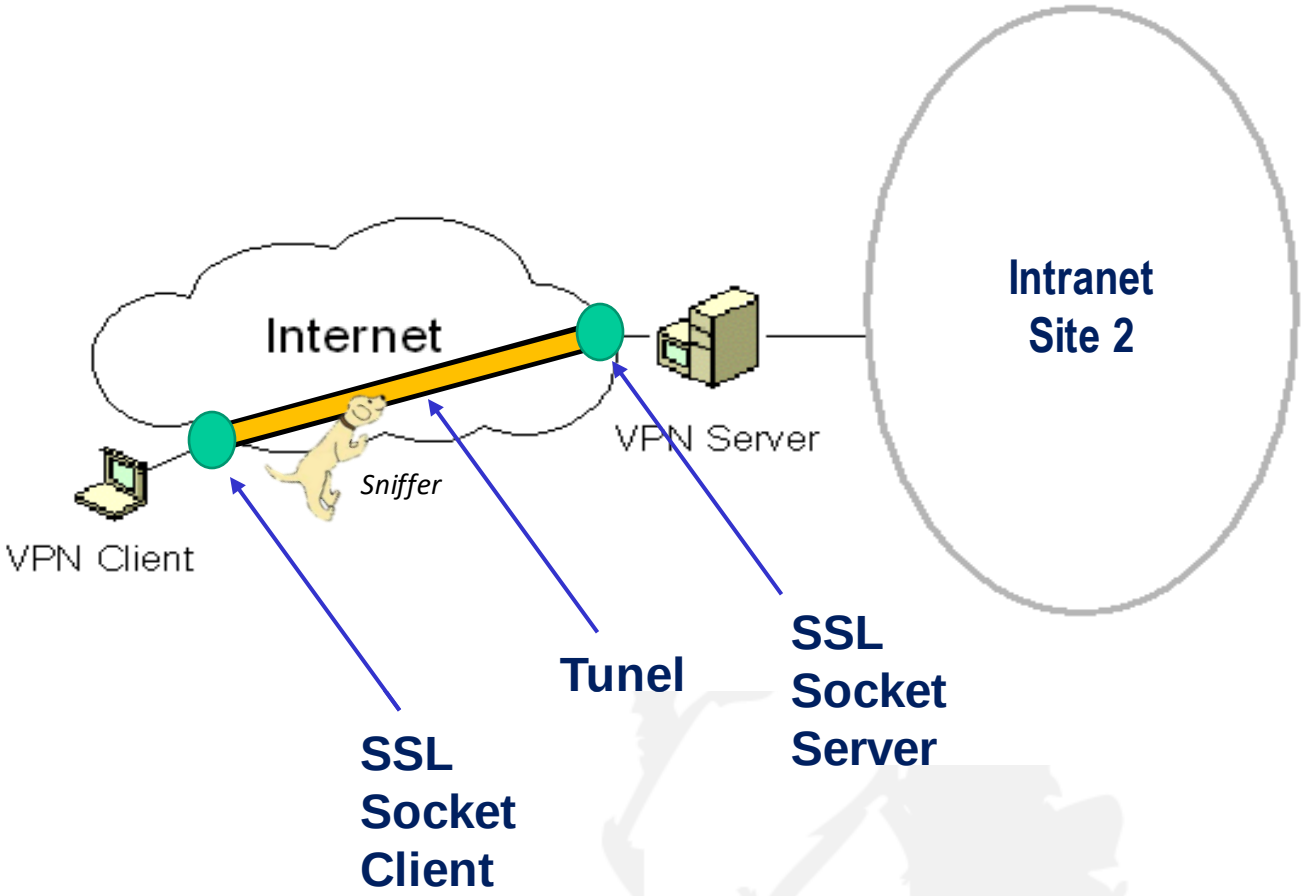
*" **Control**: The information system protects the confidentiality of transmitted information."*

VPN (Virtual Private Networks)



Tomado de GUIDE TO SSL VPNs del NIST (2008)
https://csrc.nist.gov/publications/detail/sp/800-113/final

Canal seguro – VPN SSL



Objetivos

- Desarrollar/seleccionar **cómo llevar a la práctica de forma lo más eficiente posible los canales de comunicación segura para la transmisión *de credenciales (usuario, contraseñas) y un mensaje*** con el Protocolo SSL/TLS (autenticidad, confidencialidad e integridad). Tener en cuenta que **el número de empleados que usarán la aplicación son aproximadamente 300.**

- Utilizar alguna herramienta de análisis de tráfico que **permita comprobar la confidencialidad e integridad de los canales de comunicaciones seguros**.

- Establecer los ***Cipher Suites*** que serán usados en la versión TLS 1.3. Además, **el cliente nos solicita pruebas sobre la capacidad para soportar a los 300 empleados por la VPN SSL desarrollada**.

Metricas

Resumen (30%)

- Tamaño del informe
- Calidad del resumen aportado
- Calidad de pruebas presentadas y resultados

Solución aportada (70%)

- Cumplimiento de requisitos establecidos
- Calidad del código entregado
- Complejidad de la solución
- Eficiencia de la solución
- Respuesta al conjunto de preguntas planteadas
- Pruebas realizadas

Extras de Productividad (20%):

- Aquellos ST que puedan mostrar en las sesiones de seguimiento del PAI el funcionamiento de la comunicación cliente/servidor sin estar ambos en la misma máquina +10%.
- Aquellos ST que muestren en las sesiones de seguimiento del PAI que el análisis del tráfico de red se realiza usando una tercera máquina +10%

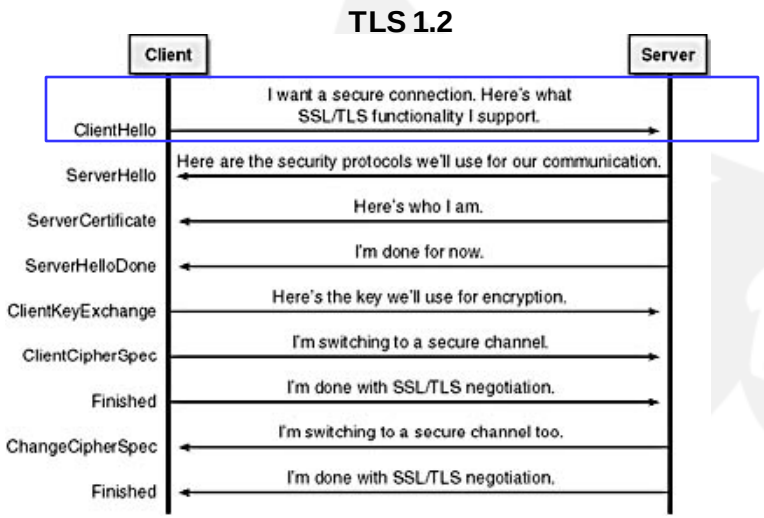
Secure Sockets Layer (SSL) / Transport Layer Security (TLS)

Protocolo de comunicación de nivel de aplicación basado en infraestructura de clave pública-privada (certificados) que permite asegurar la confidencialidad e integridad de la información, así como la autenticación de la misma:

1) **Autenticación**: Intercambio de claves/certificados
2) **Confidencialidad**: Cifrado de información
3) **Integridad**: Verificador de integridad

Protocol
SSL 1.0
SSL 2.0
SSL 3.0
TLS 1.0
TLS 1.1
TLS 1.2
TLS 1.3

1.- Handshake / CipherSuites

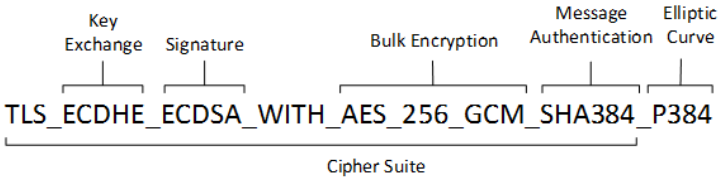


1. Cliente dice Hola!, esto son los CipherSuite que soporto

```
PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.0:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (secp256r1) - C
|       TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 2048) - C
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       64-bit block cipher 3DES vulnerable to SWEET32 attack
|   TLSv1.1:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
```

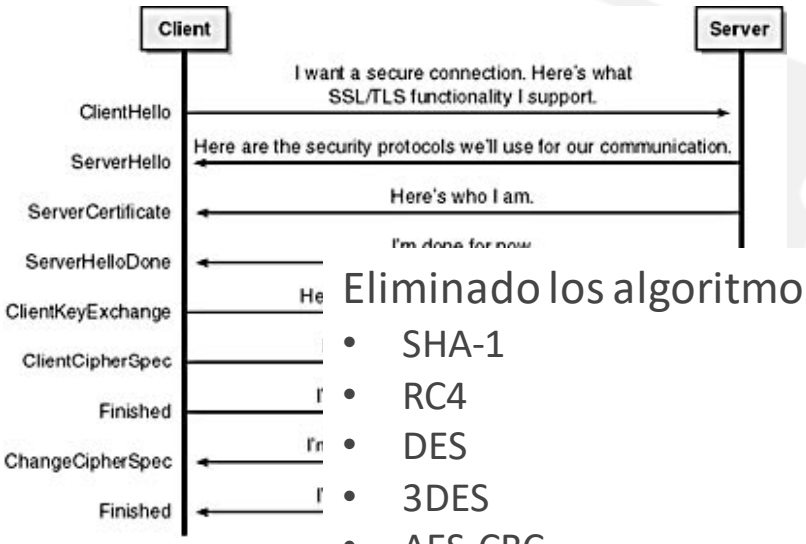
Tip: Determinar CiperSuite de un dominio
- nmap --script ssl-enum-ciphers -p 443 DOMINIO_A_ESCANEAR

Tip: Determinar CipherSuite soportadas por navegador
- https://cc.dcsec.uni-hannover.de/
- https://www.howsmyssl.com/

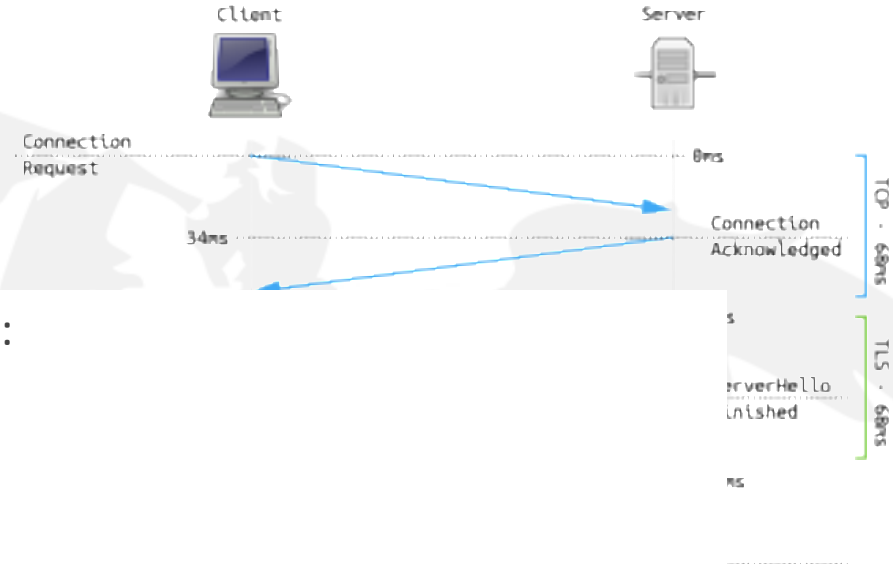


1.- Handshake / CipherSuites

TLS 1.2



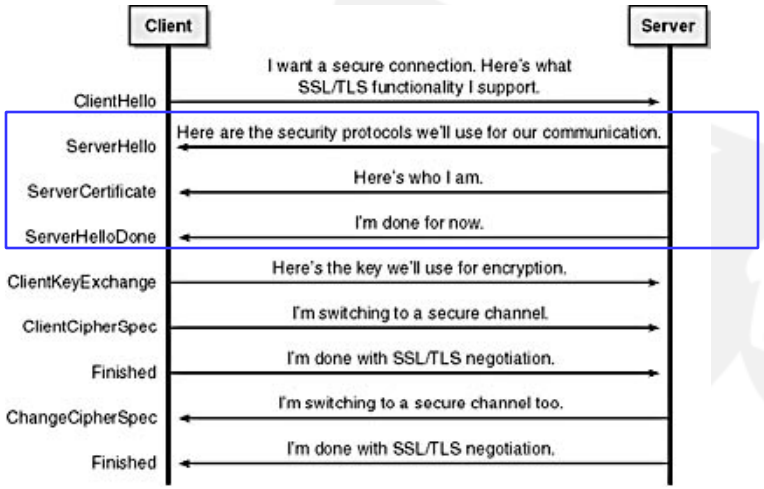
TLS 1.3



Eliminado los algoritmos:

- SHA-1
- RC4
- DES
- 3DES
- AES-CBC
- MD5
- Arbitrary Diffie-Hellman groups — CVE-2016-0701
- EXPORT-strength ciphers – Responsible for FREAK and LogJam

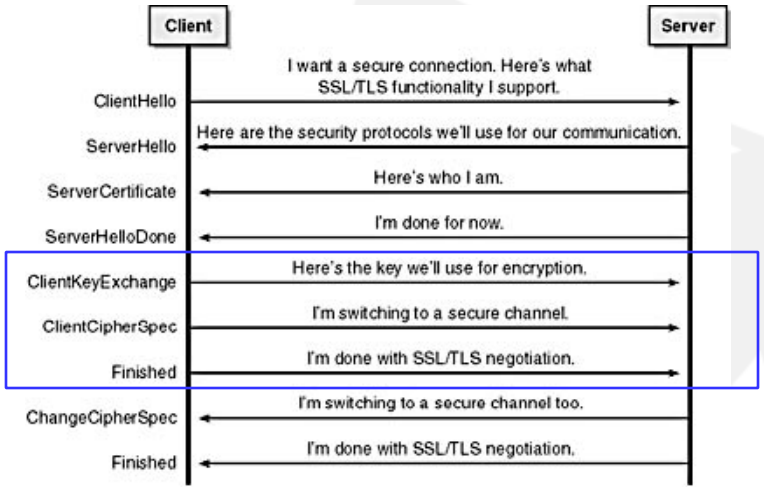
1.- Handshake / CipherSuites



2. Servidor devuelve el Hola!, vamos a usar esta CipherSuite, y te envio mi certificado, Servidor dice HECHO!:

```
PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.0:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (secp256r1) - C
|       TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 2048) - C
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       64-bit block cipher 3DES vulnerable to SWEET32 attack
|   TLSv1.1:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
```

1.- Handshake / CipherSuites

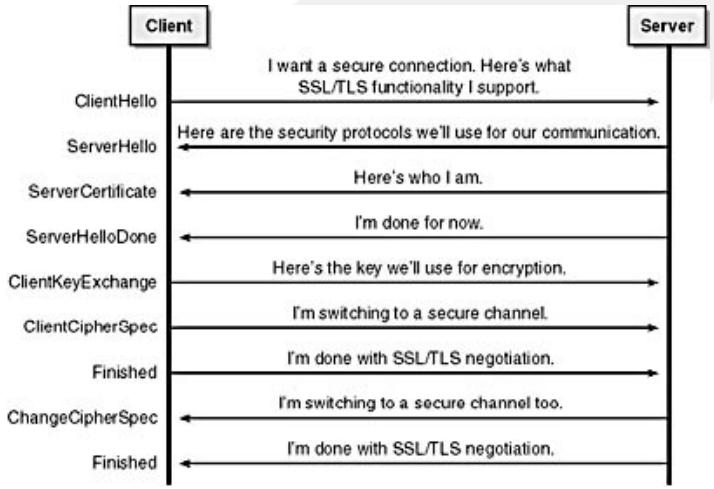


3. Cliente genera clave secreta, ésta se cifra con la clave publica del servidor que ya ha recibido, y se la envío al servidor
4. Indico al servidor que ajuste la especificación del cipher suite con esta clave.
5. Indicamos al servidor que ya hemos terminado.

FIN DEL HANDSHAKE! Just PLAY!

Key exchange / Key agreement

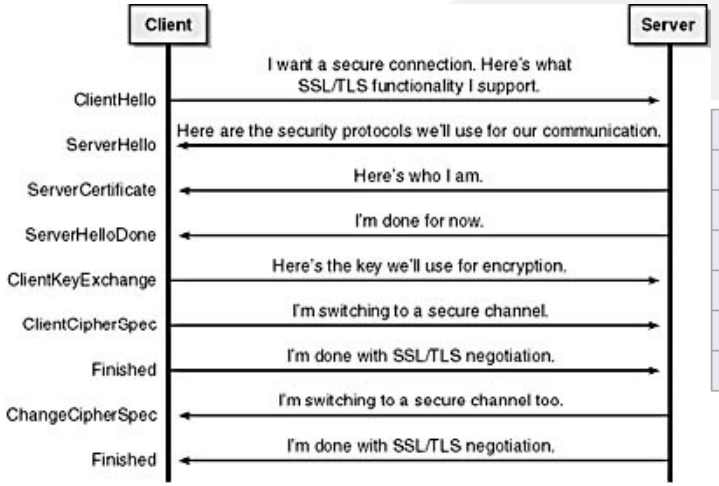
Para cumplimentar el requisitos de ***Autenticación*** debemos **negociar** el mecanismo de intercambio (key exchange) de claves, ***!Eh, compartimos claves¡***



Algorithm	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3	Status
RSA	Yes	Yes	Yes	Yes	Yes	No	
DH-RSA	No	Yes	Yes	Yes	Yes	No	
DHE-RSA (forward secrecy)	No	Yes	Yes	Yes	Yes	Yes	
ECDH-RSA	No	No	Yes	Yes	Yes	No	
ECDHE-RSA (forward secrecy)	No	No	Yes	Yes	Yes	Yes	
DH-DSS	No	Yes	Yes	Yes	Yes	No	
DHE-DSS (forward secrecy)	No	Yes	Yes	Yes	Yes	No[48]	
ECDH-ECDSA	No	No	Yes	Yes	Yes	No	
ECDHE-ECDSA (forward secrecy)	No	No	Yes	Yes	Yes	Yes	
PSK	No	No	Yes	Yes	Yes		Defined for TLS 1.2 in RFCs
PSK-RSA	No	No	Yes	Yes	Yes		
DHE-PSK (forward secrecy)	No	No	Yes	Yes	Yes		
ECDHE-PSK (forward secrecy)	No	No	Yes	Yes	Yes		
SRP	No	No	Yes	Yes	Yes		
SRP-DSS	No	No	Yes	Yes	Yes		
SRP-RSA	No	No	Yes	Yes	Yes		
Kerberos	No	No	Yes	Yes	Yes		
DH-ANON (insecure)	No	Yes	Yes	Yes	Yes		
ECDH-ANON (insecure)	No	No	Yes	Yes	Yes		
GOST R 34.10-94 / 34.10-2001[49]	No	No	Yes	Yes	Yes		Proposed in RFC drafts

Data integrity / Message Authentication Code

Para cumplimentar el requisitos de **Integridad** debemos incluir mecanismos para la comprobación de la integridad de información, ***!Eh, la información que te envío es correcta y completa¡***



Algorithm	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3	Status
HMAC-MD5	Yes	Yes	Yes	Yes	Yes	No	Defined for TLS 1.2 in RFCs
HMAC-SHA1	No	Yes	Yes	Yes	Yes	No	
HMAC-SHA256/384	No	No	No	No	Yes	No	
AEAD	No	No	No	No	Yes	Yes	
GOST 28147-89 IMIT[49]	No	No	Yes	Yes	Yes		Proposed in RFC drafts
GOST R 34.11-94[49]	No	No	Yes	Yes	Yes		

Encryption

Pra cumplimentar el requisitos de **Confidencialidad** debemos **cifrar** la información, ***!Eh, no vas a saber lo que te envío¡***

Cipher			Protocol version						Status
Type	Algorithm	Nominal strength (bits)	SSL 2.0	SSL 3.0 [n 1][n 2][n 3][n 4]	TLS 1.0 [n 1][n 3]	TLS 1.1 [n 1]	TLS 1.2 [n 1]	TLS 1.3	
Block cipher with mode of operation	AES GCM[50][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	AES CCM[51][n 5]		N/A	N/A	N/A	N/A	Secure	Secure	
	AES CBC[n 6]		N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	Camellia GCM[52][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	N/A	
	Camellia CBC[53][n 6]		N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	ARIA GCM[54][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	N/A	
	ARIA CBC[54][n 6]		N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	SEED CBC[55][n 6]	128	N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	3DES EDE CBC[n 6][n 7]	112[n 8]	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	
	GOST 28147-89 CNT[49][n 7]	256	N/A	N/A	Insecure	Insecure	Insecure	N/A	Defined in RFC 4357
	IDEA CBC[n 6][n 7][n 9]	128	Insecure	Insecure	Insecure	Insecure	N/A	N/A	Removed from TLS 1.2
	DES CBC[n 6][n 7][n 9]	56	Insecure	Insecure	Insecure	Insecure	N/A	N/A	
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	Forbidden in TLS 1.1 and later
	RC2 CBC[n 6][n 7]	40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
Stream cipher	ChaCha20-Poly1305[60][n 5]	256	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	RC4[n 11]	128	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	Prohibited in all versions of TLS by RFC 7465
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
None	Null[n 12]	–	N/A	Insecure	Insecure	Insecure	Insecure	N/A	Defined for TLS 1.2 in RFCs

Implementaciones

Implementation	SSL 2.0 (insecure)	SSL 3.0 (insecure)	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
Botan	No	No[192]	Yes	Yes	Yes	
cryptlib	No	Disabled by default at compile time	Yes	Yes	Yes	
GnuTLS	No[a]	Disabled by default[193]	Yes	Yes	Yes	yes (draft version)[194]
Java Secure Socket Extension	No[a]	Disabled by default[195]	Yes	Yes	Yes	Yes
LibreSSL	No[196]	No[197]	Yes	Yes	Yes	
MatrixSSL	No	Disabled by default at compile time[198]	Yes	Yes	Yes	yes (draft version)
mbed TLS (previously PolarSSL)	No	Disabled by default[199]	Yes	Yes	Yes	
Network Security Services	No[b]	Disabled by default[200]	Yes	Yes[201]	Yes[202]	Yes[203]
OpenSSL	No[204]	Enabled by default	Yes	Yes[205]	Yes[205]	Yes[206]
RSA BSAFE Micro Edition Suite	No	Disabled by default	Yes	Yes	Yes	Not yet
RSA BSAFE SSL-J	No	Disabled by default	Yes	Yes	Yes	Not yet
SChannel XP / 2003[207]	Disabled by default by MSIE 7	Enabled by default	Enabled by default by MSIE 7	No	No	No
SChannel Vista[208]	Disabled by default	Enabled by default	Yes	No	No	No
SChannel 2008[208]	Disabled by default	Enabled by default	Yes	Disabled by default (KB4019276)[142]	Disabled by default (KB4019276)[142]	No
SChannel 7 / 2008 R2[209]	Disabled by default	Disabled by default in MSIE 11	Yes	Enabled by default by MSIE 11	Enabled by default by MSIE 11	No
SChannel 8 / 2012[209]	Disabled by default	Enabled by default	Yes	Disabled by default	Disabled by default	No
SChannel 8.1 / 2012 R2, 10 v1507 & v1511[209]	Disabled by default	Disabled by default in MSIE 11	Yes	Yes	Yes	No
SChannel 10 v1607 / 2016[152]	No	Disabled by default	Yes	Yes	Yes	No
Secure Transport OS X 10.2–10.8 / iOS 1–4	Yes	Yes	Yes	No	No	
Secure Transport OS X 10.9–10.10 / iOS 5–8	No[c]	Yes	Yes	Yes[c]	Yes[c]	
Secure Transport OS X 10.11 / iOS 9	No	No[c]	Yes	Yes	Yes	
Seed7 TLS/SSL Library	No	Yes	Yes	Yes	Yes	
wolfSSL (previously CyaSSL)	No	Disabled by default[210]	Yes	Yes	Yes	yes (draft version)[211]
Implementation	SSL 2.0 (insecure)	SSL 3.0 (insecure)	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3